**Программа учебной дисциплины «Основы и методология программирования»**

Утверждена

Академическим советом ООП

Протокол № от «\_\_»\_\_\_\_\_\_\_ г.

|  |  |
| --- | --- |
| Автор  | Зобнин Алексей Игоревич |
| Число кредитов  | 9 |
| Контактная работа (час.)  | 342 |
| Самостоятельная работа (час.)  | 198 |
| Курс  | 1 |
| Формат изучения дисциплины | Без использования онлайн курса |

1. **ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

Цели освоения дисциплины «Основы и методология программирования» – обучить студентов навыкам программирования на языках Python и C++, а также базовым алгоритмам, необходимым в дальнейшем обучении и в работе по специальности.

В результате освоения дисциплины студент должен:

**знать:**

- основные конструкции языка программирования Python;

- основные конструкции языка программирования С++;

- основные классы и функции стандартных библиотек этих языков;

**уметь:**

- пользоваться интерпретатором языка Python и компиляторами языка C++;

- создавать программы, решающие задачи по заданному алгоритму на языках Python и С++;

- отлаживать программы;

**владеть:**

- навыками формализации и решения практических задач по программированию

- навыками работы с документацией по языку программирования.

Изучение дисциплины «Основы и методология программирования» базируется на следующих дисциплинах:

- информатика и ИКТ в объеме средней школы;

- математика в объеме средней школы.

Для освоения учебной дисциплины студенты должны владеть следующими знаниями и компетенциями:

- базовые навыки программирования (на каком-либо языке);

- умение формализовать условие задачи по программированию;

- умение применять язык программирования в новых ситуациях;

- способность записать и выполнить программу на компьютере на требуемых языках программирования;

- умение пользоваться документацией.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

- Алгоритмы и структуры данных.

1. **СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**I. Язык программирования Python.**

1. Целые числа и ввод-вывод.
2. Условный оператор.
3. Цикл while.
4. Вещественные числа и строки.
5. Функции и рекурсия.
6. Цикл for.
7. Списки.
8. Линейный поиск и сортировка.
9. Множества.
10. Словари.
11. Функциональное программирование.
12. Классы.
13. Знакомство со стандартной библиотекой языка Python.

**II. Язык программирования C++.**

1. Переменные, ввод и вывод, условный оператор, циклы, оператор switch.
2. Векторы и строки. Алгоритм sort.
3. Структуры. Классы pair и tuple.
4. Константность. Ссылки. Указатели.
5. Функции и лямбда-функции. Способы передачи и возврата параметров.
6. Ассоциативные контейнеры map и unordered\_map.
7. Контейнеры list и deque. Адаптеры stack, queue и priority\_queue.
8. Шаблонные функции. Алгоритмы стандартной библиотеки.
9. Битовые операции. Выражения, операторы. Перегрузка операторов.
10. Классы: публичный интерфейс и приватные детали реализации. Функции-члены.
11. Жизненный цикл объекта. Конструкторы, деструктор. Семантика перемещения.
12. Исключения.
13. Динамическая память. Динамические массивы. Null-terminated strings.
14. Работа с памятью. Сериализация и десериализация.
15. Идиома RAII.
16. Умные указатели unique\_ptr и shared\_ptr.
17. Одиночное наследование. Полиморфизм. Виртуальные функции.
18. Реализация шаблонного класса vector.
19. **ОЦЕНИВАНИЕ**

Текущий контроль – 3 домашних задания, а также 3 контрольных работы.

Итоговый контроль – экзамен.

Итоговая оценка вычисляется следующим образом:

*0,24 \* средняя оценка за домашние задания + 0,36 \* средняя оценка за контрольные работы + 0,4 \* оценка за экзамен*

**Таблица соответствия оценок по десятибалльной и пятибалльной системе**

|  |  |
| --- | --- |
| **По десятибалльной шкале** | **По пятибалльной системе** |
| 1 – неудовлетворительно 2 – очень плохо3 – плохо | неудовлетворительно – 2 |
| 1. – удовлетворительно
2. – весьма удовлетворительно
 | удовлетворительно – 3 |
| 1. – хорошо
2. – очень хорошо
 | хорошо – 4 |
| 8 – почти отлично 9 – отлично10 - блестяще | отлично – 5 |

1. **ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ**

## **Примеры домашних задач**

1. Напишите программу, составляющую предметный указатель для ключевых слов книги.
2. Напишите свои версии следующих утилит Unix: wc, rev, cat, sort, uniq.
3. Реализуйте шаблонный класс «Многочлен от одной переменной».
4. Создайте свою реализацию шаблонного адаптера stack на C++.
5. Напишите собственную версию «умного» указателя shared\_ptr на C++.
6. Отрецензируйте данный код, указав и исправив в нём все ошибки.
7. Напишите функцию-фабрику для полиморфной работы с заданной иерархией классов.

## **Примеры заданий контрольных работ**

1. Напишите функцию, вычисляющую длину максимального общего префикса двух строк.
2. Напишите функцию, определяющую, является ли данная строка палиндромом.
3. Напишите программу, которая для каждого слова из текстового файла подсчитывала бы количество его вхождений.
4. Напишите функцию, которая бы выводила на экран битовое представление числа.
5. Напишите функцию, которая проверяла бы, что строка, состоящая из открывающих и закрывающих скобок, корректно сформирована.
6. Напишите собственную реализацию шаблонных функций copy\_if, rotate и unique.
7. Реализуйте класс «Рациональное число».

## **Примеры заданий экзамена**

1. Напишите код, удаляющий из данного контейнера все нули.
2. Сколько байт памяти занимает типичная реализация контейнера std::vector<T> на стеке?
3. Какие контейнеры стандартной библиотеки С++ допускают обращение по индексу?
4. Чем отличаются контейнеры map и unordered\_map в C++?
5. Что делает функция стандартной библиотеки C++ remove?
6. Зачем могут потребоваться приватные поля и функции в классе?
7. Почему не следует генерировать исключения в деструкторах классов?

1. **РЕСУРСЫ**
	1. **Основная литература**

1. Страуструп, Б. Язык программирования C++ / Б. Страуструп; Пер. с англ. под ред. Н. Н. Мартынова. – Спец. изд. – М.: БИНОМ, 2012. – 1135 с. (или более позднее издание).

 **Дополнительная литература**

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 1999. – 960 с. (или более позднее издание).

* 1. **Программное обеспечение**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
|   1. | Microsoft Windows 7 Professional RUSMicrosoft Windows 10Microsoft Windows 8.1 Professional RUS | *Из внутренней сети университета (договор)* |
| 2. |  Ubuntu 14.04 или выше | *Свободно распространяемая системаURL: http://releases.ubuntu.com/* |
| 3. | Интерпретатор языка Python 3.x  | *Свободно распространяемое программное обеспечение* |
| 4. | Компилятор языка C++11 (g++ 4.8) | *Свободно распространяемое программное обеспечение* |

* 1. **Профессиональные базы данных, информационные справочные системы,
	интернет-ресурсы (электронные образовательные ресурсы)**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
|  | ***Профессиональные базы данных, информационно-справочные системы*** |
| 1. | Документация по языку Pyhton | *URL: https://docs.python.org/3/* |
| 2. | Документация по языку C++ | *URL: https://en.cppreference.com/* |
|  | ***Интернет-ресурсы (электронные образовательные ресурсы)*** |
| 1. | Сервис для онлайн-проверки заданий Яндекс.Контест | *URL: https://contest.yandex.ru/* |

* 1. **Материально-техническое обеспечение дисциплины**

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

* ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);
* мультимедийный проектор с дистанционным управлением.

Учебные аудитории для лабораторных и самостоятельных занятий по дисциплине оснащены персональными компьютерами (ноутбуками) с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде НИУ ВШЭ.